



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

AF/\$
2700

Re application of: **Bennett**

Serial No.: **08/797,079**

Filed: **February 10, 1997**

For: **Method for File Transfer
Restarts Using Standard Internet
Protocol**

§ Group Art Unit: **2153**
§
§ Examiner: **Dinh, Dung C.**
§ Attorney Docket No.: **AT9-97-044**

SA-27

Certificate of Mailing Under 37 C.F.R. § 1.8(a)
I hereby certify this correspondence is being deposited with the United States Postal Service as First Class mail in an envelope addressed to: Assistant Commissioner of Patents, Washington, D.C. 20231 on September 18, 2002.
By: Dell Whitton
Dell Whitton

TRANSMITTAL DOCUMENT

Assistant Commissioner of Patents
Washington, D.C. 20231

RECEIVED

OCT 01 2002

Technology Center 2100

Sir:
ENCLOSED HEREWITH:

- Appellant's Brief (in triplicate) (37 C.F.R. 1.192); and
- Our return postcard.

A fee of \$320.00 is required for filing an Appellant's Brief. Please charge this fee to IBM Deposit Account No. 09-0447. No additional fees are believed to be necessary. If, however, any additional fees are required, I authorize the Commissioner to charge these fees which may be required to Deposit Account No. 09-0447. No extension of time is believed to be necessary. If, however, an extension of time is required, the extension is requested, and I authorize the Commissioner to charge any fees for this extension to Deposit Account No. 09-0447.

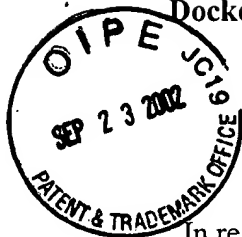
Respectfully submitted,

Duke W. Yee
Duke W. Yee
Registration No. 34,285
CARSTENS, YEE & CAHOON, LLP
P.O. Box 802334
Dallas, Texas 75380
(972) 367-2001
ATTORNEY FOR APPLICANT

#22

Docket No. AT9-97-044

PATENT



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Bennett**

Serial No.: **08/797,079**

Filed: **February 10, 1997**

For: **Method for File Transfer Restarts
Using Standard Internet Protocol**

§
§
§
§
§
§
§

Group Art Unit: **2153**

Examiner: **Dinh, Dung C.**

RECEIVED

OCT 01 2002

Technology Center 2100

Assistant Commissioner for Patents
Washington, D.C. 20231

**ATTENTION: Board of Patent Appeals
and Interferences**

Certificate of Mailing Under 37 C.F.R. § 1.8(a)

I hereby certify this correspondence is being deposited with the United States Postal Service as First Class mail in an envelope addressed to: Assistant Commissioner of Patents, Washington, D.C. 20231 on September 18, 2002.

By:

Dell Whitten
Dell Whitten

APPELLANT'S BRIEF (37 C.F.R. 1.192)

This brief is in furtherance of the Notice of Appeal, filed in this case on July 19, 2002.

The fees required under § 1.17(c), and any required petition for extension of time for filing this brief and fees therefore, are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

This brief is transmitted in triplicate. (37 C.F.R. 1.192(a))

09/25/2002 AWONDAF1 00000042 090447 08797079

01 FC:120 320.00 CH

REAL PARTIES IN INTEREST

The real party in interest in this appeal is the following party: IBM Corporation

RELATED APPEALS AND INTERFERENCES

With respect to other appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interferences.

STATUS OF CLAIMS

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

Claims in the application are: 1, 3-10, 12, 14-17, 21-25, and 31-39

B. STATUS OF ALL THE CLAIMS IN APPLICATION

1. Claims canceled: 2, 11, 13, 18-20, and 26-30
2. Claims withdrawn from consideration but not canceled: NONE
3. Claims pending: 1, 3-10, 12, 14-17, 21-25, and 31-39
4. Claims allowed: NONE
5. Claims rejected: 1, 3-10, 12, 14-17, 21-25, and 31-39

C. CLAIMS ON APPEAL

The claims on appeal are: 1, 3-10, 12, 14-17, 21-25, and 31-39

STATUS OF AMENDMENTS

There are no amendments after final rejection.

SUMMARY OF INVENTION

The present invention provides a file transfer mechanism that uses conventional protocols and allows for file transfer restarts, particularly with large files. A large file is broken up into discrete pieces, each piece being a component file. A profile includes a master list of the component files. See specification, page 17, lines 12, to page 19, line 3. When a download is initiated, the profile is retrieved. Each component file is downloaded individually. When all component files are successfully downloaded, the original file is reconstructed from the component files. See specification, page 20, line 6, to page 22, line 9. If the file transfer is interrupted, the server may re-send the profile and the client may check for existing component files and restart transmission with the first missing component file. See specification, page 22, line 10, to page 23, line 7.

ISSUES

The issues on appeal are as follows:

Whether claims 1, 3-5, 8, 9, 12, 14-17, 21-25, and 31-39 are unpatentable as being obvious over Kauffman et al. (US Patent No. 5,857,203) in view of Averbuch et al. (US Patent No. 5,689,825).

Whether claims 6, 7, 10, and 16 are unpatentable as being obvious over Kauffman et al. in view of Averbuch et al. and further in view of Pyne (US Patent No. 5,446,888).

GROUPING OF CLAIMS

The claims on appeal do not stand or fall in a single group, but are grouped into the following groups:

Claims 1, 4, 5, 8, 9, 12, 14, 15, 17, 21-25, and 31 form group A. Claim 3 forms group B. Claims 6, 7, 10, and 16 form group C. Claims 32-39 form group D.

ARGUMENT

The Office Action rejects claims 1, 3-5, 8-9, 12, 14-17, 21-25, and 31-39 under 35 U.S.C. § 103 as being unpatentable over Kauffman et al. (US Patent No. 5,857,203), hereinafter referred to as “*Kauffman*” in view of Averbuch et al. (US Patent No. 5,689,825), hereinafter referred to as “*Averbuch*.” This rejection is respectfully traversed.

I. The Prior Art Fails to Teach or Suggest the Claimed Invention (Group A)

With respect to claim 1, the Office Action states:

As per claim 1, Kauffman teaches a method of downloading a file, consisting of components, from a server to a client, comprising the steps of: receiving from a server a profile [piece map] of the download file; initiating a download sequence by which each component file is transferred and reassembling the component files into the download file using the profile [col.11 lines 14-29].

Kauffman does not specifically disclose the client and server being and Internet client and an Internet server using an Internet protocol. However, applicant defines (on page 25 of the specification) an Internet client and Internet server as any type of computers or components connected to any type of network. Hence, the client and server of Kauffman read on the Internet client, Internet server as claimed.

It would have been obvious for one of the ordinary skill in the art to use Internet protocol (e.g. FTP) because it is widely available, standard reliable protocol for transferring files.

Kauffman does not specifically disclose what happened when download is interrupted. Averbuch teaches a method for transferring file sequences wherein upon interruption of the download sequence, restarting the download at the component affected by the interruption [col.6 lines 20-29]. The component transferred prior to the interruption is not re-transferred [col.6 line 29]. Hence, it would have been obvious for one of ordinary skill in the art to combine the teaching of Averbuch with the system of Kauffman because it would have improve the efficiency of the downloading.

Office Action, dated 10 October 2001. Appellant respectfully disagrees. *Kauffman* teaches an object library in which large files are divided into smaller pieces, which are themselves files, and creates a piece map identifying those pieces. *Kauffman*, col. 6, lines 34-39. When a client performs a retrieve function on one of the large files, the server downloads the piece map. The client then decodes the piece map and downloads the individual pieces to reconstitute the large file. *Kauffman*, col. 10, line 58, to col. 11, line 28. *Kauffman* does not teach or suggest, when a

download sequence is interrupted, restarting the download sequence with the piece affected by the interruption, as specifically recited in claim 1.

Averbuch teaches a method and apparatus for downloading updated software to portable wireless communication units. The method of *Averbuch* determines download session parameters including a block size and a number of blocks to be transferred. These parameters are downloaded to and stored in the client. *Averbuch* states, “[w]ith knowledge of the size of the original data transfer, the block size, and the remaining number of blocks to be transferred, the data transfer may continue from where it was left off.” *Averbuch*, col. 6, lines 24-27. *Averbuch* teaches continuing data transfer of a single file if the transfer is interrupted. However, *Averbuch* provides no teaching or suggestion regarding downloading a plurality of piece files. At best, *Averbuch* teaches what to do if data transfer of a single piece file is interrupted.

Furthermore, *Averbuch* actually teaches away from the presently claimed invention since *Averbuch* directs one to downloading blocks with a fixed block size rather than dividing the file into pieces, as in the claimed invention. See *In re Hedges*, 228 U.S.P.Q. 685 (Fed. Cir. 1986). Thus, one of ordinary skill in the art would not be motivated to make the changes proposed by the Office Action.

The present invention recognizes the disadvantage of using file transfer protocol (FTP) for download of large files using a device that is likely to be interrupted during download. *Kauffman* does not teach the problem or its source. Instead, *Kauffman* is directed towards storage of large files in particular file systems. However, *Averbuch* does teach the problem, but offers a very different solution. *Averbuch* actually teaches transferring the file in fixed data blocks and tracking the number of blocks remaining. This solution is very different from the invention recited in claim 1 and would not lead a person of ordinary skill in the art to make the proposed combination. Therefore, one of ordinary skill in the art would not be motivated to combine or modify the references in the manner required to form the solution disclosed in the claimed invention.

The present invention is directed towards dividing a download file into a plurality of component files and, when a download sequence is interrupted, restarting the download sequence with the piece affected by the interruption. Even if *Kauffman* and *Averbuch* could be properly combined, the combination would not form the invention recited in claim 1. Instead, a combination of *Kauffman* and *Averbuch* would result in a method for downloading a file, in

which the file is divided into a plurality of piece files that, once downloaded, may be used to reconstitute the document file and, when transfer of one of the piece files is interrupted, the number of blocks remaining is used to restart the transfer of the piece file.

The Final Office Action states:

Applicant argued that the present invention is “directed towards dividing file into plurality of component files and, when a download sequence is interrupted, restarting the download with the piece affected by the interruption”. Applicant further argued that the combination of Kauffman and Averbuch would result in “a method for downloading a file, in which the file is divided into a plurality of piece files ..., when transfer of one of the piece files is interrupted, the number of blocks remaining is used to restart the transfer of the piece file.”

The examiner failed to see how this reading of the combination is distinguished from the claimed invention. Clearly, even as argued by Applicant, the combination teaches dividing a file into piece files, restarting download of the piece file that was interrupted, and combining the piece files after downloading to reconstruct the file. There is no rationale in Kauffman or Averbuch to support applicant argument that the combination would result in Kauffman restarting the download by transferring every piece files including the one already downloaded.

Final Office Action, dated May 1, 2002. Appellant respectfully disagrees. Appellant notes that there is no suggestion in *Kauffman* or *Averbuch* to support the combination. Even assuming, *arguendo*, that a person of ordinary skill in the art would combine *Kauffman* and *Averbuch*, neither reference teaches how the two very different approaches may be combined. In response to the statement, “[t]here is no rationale in Kauffman or Averbuch to support applicant argument that the combination would result in Kauffman restarting the download by transferring every piece files including the one already download,” appellant notes that there is no rationale for the combination in the first place. It follows that such a combination could take many forms, assuming someone would be so motivated to force the combination. For example, if one were to somehow combine *Kauffman* and *Averbuch*, perhaps the file would be divided into a plurality of component files of a fixed size and the resulting combination would count the number of component files, rendering the piece map useless. Perhaps a single file would be associated with a file map that would be used to download blocks of a fixed size. However, neither these examples nor any others are actually supported by the references themselves. There is no suggestion to combine *Kauffman* and *Averbuch* and, thus, there is no teaching as to how the two very different approaches would be combined.

Furthermore, *Kauffman* teaches dividing a large file into smaller component files. *Averbuch* teaches counting blocks of a fixed size, which is the opposite approach to that of *Kauffman*. Since the approaches taken in *Kauffman* and *Averbuch* are actually mutually opposed to one another, neither reference suggest its combination with the other. In addition, there is no reasonable expectation that the combination could be successful. To combine these very different approaches would require undue experimentation, assuming someone would be so motivated to force the combination. Hence, the references, when considered as a whole, would not enable the combination. Therefore, one of ordinary skill in the art would not be motivated to make the proposed combination to reach the presently claimed invention when *Kauffman* and *Averbuch* are considered as a whole. The many possible results of the combination are irrelevant because a person of ordinary skill in the art would not be motivated to make the proposed combination.

Moreover, the Office Action may not use the claimed invention as an "instruction manual" or "template" to piece together the teachings of the prior art so that the invention is rendered obvious. *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992). Such reliance is an impermissible use of hindsight with the benefit of applicant's disclosure. *Id.* Therefore, absent some teaching, suggestion, or incentive in the prior art, *Kauffman* and *Averbuch* cannot be properly combined to form the claimed invention. The presently claimed invention can be reached only through the an impermissible use of hindsight with the benefit of appellant's disclosure a model for the needed changes.

The Final Office Action states:

In response to applicant's argument that the examiner's conclusion of obviousness is based upon improper hindsight reasoning, it must be recognized that any judgment on obviousness is in a sense necessarily a reconstruction based upon hindsight reasoning. But so long as it takes into account only knowledge which was within the level of ordinary skill at the time the claimed invention was made, and does not include knowledge gleaned only from the applicant's disclosure, such reconstruction is proper. See *In re McLaughlin*, 443 F.2d 1392, 170 USPQ 209 (CCPA 1971).

Final Office Action, dated May 1, 2002. Appellant respectfully disagrees. The Office Action introduces absolutely no "knowledge which was within the level of ordinary skill at the time the claimed invention was made" that supports or suggests the combination. The Office Action proposes combining *Kauffman* and *Averbuch* "because it would have improve the efficiency of

the downloading.” See Office Action, dated October 10, 2001. However, *Kauffman* does not teach or suggest that the efficiency of downloading piece files could be improved by counting fixed-sized blocks, as in *Averbuch*. And *Averbuch* does not teach or suggest that efficiency of downloading a file could be improved by dividing a file into piece files, rather than using fixed-sized blocks. Since *Kauffman* and *Averbuch* offer very different approaches to downloading files and provide no incentive to combine their teachings, the combination can only be made using “knowledge gleaned only from the applicant’s disclosure.” Therefore, the reconstruction is improper.

Since claims 3-5 and 32 depend from claim 1, the same distinctions between *Kauffman* and *Averbuch* and the invention recited in claim 1 apply for these claims. Claims 8-9, 12, 14-17, 21-25, and 33-39 are allowable for the reasons stated above with respect to claim 1. Additionally, claims 3-5, 8-9, 12, 14-17, 21-25, and 33-39 recite other additional combinations of features not suggested by the reference.

II. The Prior Art Fails to Teach or Suggest Any Component File Transferred Prior to the Interruption Not Being Re-transferred from the Server (Group B)

With respect to claim 3, the Office Action states:

As per claim 3, since *Kauffman*’s download file comprises plural component files, it is inherent that only the affected component file is restarted. It would have been obvious for one of ordinary skill in the art not to re-transmit completed component files prior to the interruption so as to conserve time and bandwidth.

Office Action, dated 10 October 2001. The Office Action misapplies the concept of “inherency.”

Under the principles of inherency, a claim is anticipated if a structure in the prior art necessarily functions in accordance with the limitations of a process or method claim. *In re King*, 801 F.2d 1324, 231 U.S.P.Q. 136 (Fed. Cir. 1986). Neither *Kauffman* nor *Averbuch* teaches or suggests that, in the case of an interruption, the download starts with the affected component file. Clearly, even if one were to combine *Kauffman* and *Averbuch*, an interrupted download could restart with any component file. More likely, when a download sequence is interrupted, the client may start from scratch by downloading the piece map and downloading every piece file identified in the piece map. Regardless, the Office Action proffers no reasoning as to why, in a combination of *Kauffman* and *Averbuch*, an interrupted download must **necessarily** restart with the affected file.

The Office Action offers a motivation of conservation of time and bandwidth to support the proposed modification to the applied art. The Office Action may not make modifications to the prior art using the claimed invention as a model for the modifications. *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780, 1783-1784 (Fed. Cir. 1992). "The mere fact that the prior art may be modified in the manner suggested by the Examiner does not make the modification obvious unless the prior art has suggested the desirability of the modification." *Id.* In other words, unless some teaching exists in the prior art for the suggested modification, merely asserting that such a modification would be obvious to one of ordinary skill in the art is improper and cannot be used to meet the burden of establishing a *prima facie* cases of obviousness. Such reliance is an impermissible use of hindsight with the benefit of appellant's disclosure.

III. The Prior Art Fails to Teach or Suggest the Download Sequence Being Restarted Using the Profile (Group D)

Also, with respect to claims 32-39, the Office Action states:

As per claims 32-39, it is inherent that the system as modified would used the profile to continue the download in order to identify with component files are to be request from the server.

Office Action, dated 10 October 2001. Again, the Office Action misapplies the concept of "inherency." One can assume that, in the system of *Kauffman*, as allegedly combined with *Averbuch*, when a download sequence is interrupted, the client may start from scratch by downloading the piece map and downloading every piece file identified in the piece map. Regardless, the Office Action proffers no reasoning as to why, in a combination of *Kauffman* and *Averbuch*, an interrupted download must **necessarily** restart with the affected file or use the profile to continue the download. As stated above, *Averbuch* teaches transferring a download file in fixed data blocks, tracking the number of blocks remaining, and continuing data transfer of a single file if the transfer is interrupted. This solution is very different from the invention recited in claims 32-39, wherein the **profile** is used to restart a **download sequence**.

Therefore, the rejection of claims 1, 3-5, 8-9, 12, 14-17, 21-25, and 31-39 under 35 U.S.C. § 103 is overcome.

IV. The Prior Art Fails to Teach or Suggest File Integrity Checking in File Transfer Restarts (Group C)

The Office Action rejects claims 6-7, 10, and 16 under 35 U.S.C. § 103 as being unpatentable over *Kauffman* in view of *Averbuch* and further in view of *Pyne* (US Patent No. 5,446,888). This rejection is respectfully traversed.

With respect to claims 6-7, 10, and 16, the Office Action states:

As per claims 6-7, 10, 16, *Kauffman* does not specifically disclose using CRC for identifying file component and verifying the component file integrity. The use of CRC to identify and verify file integrity is well known in the art. *Pyne* teaches a remote file transfer method using CRC to identify and check the integrity of the file [see col.7 lines 50-64]. It would have been obvious for one of ordinary skill in the art use CRC as identifier for the component file because it would have improved the reliability of the system by enabling the client to identify and check the integrity of downloaded component files.

Office Action, dated 10 October 2001. Appellant respectfully disagrees. *Pyne* teaches the use of CRC for checking file integrity. However, the prior art fails to teach or suggest CRC codes in identifying information for each component file of a download file in a profile, which is received before initiating a download of the component files. More specifically, no suggestion to modify the combination of *Kauffman* and *Averbuch* to include such a combination of features has been shown in the Office Action. Instead, *Pyne* teaches a single file rather than a plurality of component files. Furthermore, *Pyne* teaches computing key values at the receiving computer and transferring the key values to the source computer. See col. 4, lines 41-51. Thus, *Pyne* does not teach including CRC values in a profile.

The Final Office Action states:

As per the argument concerning the *Pyne* reference, the argument is not persuasive because *Pyne* clearly teaches dividing file into “blocks” and providing a reference key (CRC) for each block [see for example *Pyne*’s claim 1 step a)].

Final Office Action, dated May 1, 2002. Appellant respectfully disagrees. A client computing a reference key for a block, as taught in *Pyne*, is not equivalent to CRC codes in identifying information for each component file of a download file in a profile, which is received before initiating a download of the component files. *Pyne* teaches computing key values at the receiving computer and transferring the key values to the source computer. See col. 4, lines 41-51. Thus, *Pyne* provides no teaching or suggestion of including CRC values in a profile. Even if

Kauffman, Averbuch, and Pyne could be somehow combined, the combination still does not teach or suggest each and every claim limitation. Therefore, claims 6-7, 10, and 16 cannot be rendered obvious by *Kauffman, Averbuch, and Pyne*.

The present invention is directed towards dividing a download file into a plurality of component files and, when a download sequence is interrupted, restarting the download sequence with the piece affected by the interruption. Even if *Kauffman, Averbuch, and Pyne* could be properly combined, the combination would not form the invention recited in claims 6-7, 10, and 16. Instead, a combination of *Kauffman, Averbuch, and Pyne* would result in a method for downloading a file, in which the file is divided into a plurality of piece files that, once downloaded, may be used to reconstitute the document file and, when transfer of one of the piece files is interrupted, the number of blocks remaining is used to restart the transfer of the piece file, and if a destination file corresponding to the source file already exists on the client computer, the client computer may download only blocks that have changed.

Therefore, the rejection of claims 6-7, 10, and 16 under 35 U.S.C. § 103 is overcome.

V. **Conclusion**

In view of the above comments, it is respectfully urged that the rejections of the claims not be sustained.



Stephen R. Tkacs
Reg. No. 46,430
Carstens, Yee & Cahoon, LLP
PO Box 802334
Dallas, TX 75380
(972) 367-2001

APPENDIX OF CLAIMS

The text of the claims involved in the appeal is:

1. A method, in an Internet client, of downloading a download file, consisting of a set of component files, from an Internet server, comprising:

receiving from the server a profile of the download file that includes identifying information for each component file;

initiating a download sequence by which each component file is transferred, one-by-one, from the server using an Internet protocol;

upon interruption of the download sequence, restarting the download sequence with a component file affected by the interruption; and

when the download sequence is complete, reassembling the component files into the download file using the identifying information in the profile.

3. The method as described in Claim 1 wherein any component file transferred prior to the interruption is not re-transferred from the server.

4. The method as described in Claim 1 wherein the Internet protocol is the File Transfer Protocol (FTP).

5. The method as described in Claim 1 wherein the identifying information in the profile for each component file includes an identifier, a value indicating a size of the component file, and a code uniquely identifying the component file.

6. The method as described in Claim 5 wherein the code is a cyclic redundancy code.

7. The method as described in Claim 5 further including verifying that a component file transferred from the server is part of the download file using the identifying information.

8. A method, in an Internet server, of downloading a download file, consisting of a set of component files, to an Internet client, comprising:

breaking the download file into the set of component files;
generating a profile of the download file that includes identifying information for each component file;

initiating a download sequence by which each component file is transferred, one-by-one, to the client using an Internet protocol; and

responsive to any interruption of the download sequence, restarting the download sequence with a component file affected by the interruption.

9. The method as described in Claim 8 wherein the identifying information in the profile for each component file includes an identifier, a value indicating a size of the component file, and a code uniquely identifying the component file.

10. The method as described in Claim 9 wherein the code is a cyclic redundancy code.

12. A method, in an Internet client, of downloading a download file, consisting of a set of component files, from an Internet server, the download file represented by a profile that includes identifying information for the download file and for each component file thereof, comprising:

transferring the profile from the server;

initiating a download sequence according to the profile by which each component file is transferred, one-by-one, from the server using Internet File Transfer Protocol (FTP);

upon receipt of a component file, using the identifying information to verify whether a complete version of the component file has been transferred;

if the complete version of the component file has not been transferred, restarting the download sequence with the component file; and

when the download sequence is complete, reassembling the component files into the download file and verifying whether a complete version of the download file has been transferred using the identifying information for the download file.

14. The method as described in Claim 12 further including re-transferring the profile from the server prior to restarting the download sequence.

15. The method as described in Claim 12 wherein the identifying information for the download file includes a code uniquely identifying the download file.
16. The method as described in Claim 15 wherein the code is a cyclic redundancy code.
17. A computer program product, in a computer readable medium, for use in an Internet client for downloading a download file, consisting of a set of component files, from a Internet server, the computer program product comprising:
- instructions for receiving from the server a profile that includes identifying information for the download file and for each component file thereof;
 - instructions for initiating a download sequence by which each component file is transferred, one-by-one, from the server using an Internet protocol;
 - instructions responsive to any interruption of the download sequence, for restarting the download sequence with the component file affected by the interruption; and
 - instructions responsive to completion of the download sequence for reassembling the component files into the download file using the identifying information in the profile.
21. A computer program product, in a computer readable medium, for use in an Internet server for downloading a download file to an Internet client, the computer program product comprising:
- instructions for breaking the download file into a set of component files;
 - instructions for generating a profile that includes identifying information for the download file and for each component file thereof;
 - instructions for transferring the profile to the client;
 - instructions for initiating a download sequence according to the profile by which each component file is transferred, one-by-one, to the client using an Internet protocol; and
 - instructions responsive to any interruption of the download sequence for restarting the download sequence with the component file affected by the interruption.
22. A client computer connectable to the Internet, comprising:
- a processor; and

a memory electrically connected to the processor, the memory having stored therein Internet protocol instructions and a file transfer download routine to be executed by the processor for performing the following steps:

- receiving from a server a profile of a download file that includes identifying information for a plurality of component files that make up the download file;
- initiating a download sequence by which each component file is transferred, one-by-one, from the server using the Internet protocol instructions;
- responsive to an interruption of the download sequence, restarting the download sequence with the component file affected by the interruption; and
- responsive to receipt of the component files, reassembling the component files into the download file using the profile.

23. The client computer as described in Claim 22 wherein the Internet protocol instructions are file transfer protocol instructions.

24. A server computer connectable to the Internet, comprising:

- a processor; and

- a memory electrically connected to the processor, the memory having stored therein Internet protocol instructions and a file transfer download routine to be executed by the processor for performing the following steps:

- breaking a download file into a set of component files;
- generating a profile of the download file that includes identifying information for each component file;
- transferring the profile to a client;
- initiating a download sequence by which the component files are transferred, one-by-one, to the client using the Internet protocol instructions; and
- responsive to any interruption of the download sequence, restarting the sequence with the component file affected by the interruption.

25. The client computer as described in Claim 24 wherein the Internet protocol instructions are file transfer protocol instructions.

31. A data processing system, comprising:
a remote control unit; and
a base unit connectable to a monitor for providing Internet access under the control of the remote control unit, the base unit comprising:
a processor; and
a memory electrically connected to the processor, the memory having stored therein Internet protocol instructions and a file transfer download routine to be executed by the processor for performing the following steps:
receiving from a server a profile of a download file that includes identifying information for a plurality of component files that make up the download file;
initiating a download sequence by which each component file is transferred, one-by-one, from the server using the Internet protocol instructions;
responsive to an interruption of the download sequence, restarting the download sequence with the component file affected by the interruption; and
responsive to receipt of the component files for reassembling the component files into the download file using the profile.
32. The method of claim 1, wherein the download sequence is restarted using the profile.
33. The method of claim 8, wherein the download sequence is restarted using the profile.
34. The method of claim 12, wherein the download sequence is restarted using the profile.
35. The computer program product of claim 17, wherein the download sequence is restarted using the profile.
36. The computer program product of claim 21, wherein the download sequence is restarted using the profile.
37. The client computer of claim 22, wherein the download sequence is restarted using the profile.

38. The server computer of claim 24, wherein the download sequence is restarted using the profile.

39. The data processing system of claim 31, wherein the download sequence is restarted using the profile.